

A Simple and Efficient Kinetic Spanner*

Mohammad Ali Abam[†] Mark de Berg[†] Joachim Gudmundsson[‡]

Abstract

We present a kinetic data structure for maintaining a $(1 + \varepsilon)$ -spanner of size $O(n/\varepsilon^2)$ for a set of n moving points in the plane. Assuming the trajectories of the points can be described by polynomials whose degrees are at most s , the number of events processed by our structure is $O((n/\varepsilon^2) \cdot \lambda_{s+2}(n))$, where $\lambda_{s+2}(n)$ denotes the maximum length of a $(n, s + 2)$ -Davenport-Schinzel sequence. Each event can be handled in $O(1)$ time, plus $O(\log n)$ time to update the event queue.

1 Introduction

A *geometric network* on a set P of n points in d -dimensional space is an undirected weighted graph $G(P, E)$ with vertex set P whose edges are straight-line segments connecting pairs of points in P . The weight of an edge (p, q) in a geometric network is equal to the distance between p and q . Often the space considered is the Euclidean plane, but other metrics and/or higher dimensions can be considered as well. Geometric networks naturally model many real-life networks, such as road networks, telecommunication networks, and so on. When designing a network for a given set P of points, several criteria can be taken into account. In particular, in many applications it is important to ensure a fast connection between every pair of points in P . For this it would be ideal to have a direct connection between every pair of points—the network would then be a complete graph—but in most applications this is unacceptable due to the high costs. This leads to the concept of spanners, as defined next.

For a geometric graph $G(P, E)$ and two points $p, q \in P$, we use $d_G(p, q)$ to denote their distance in the graph, that is, the length of the (weighted) shortest path between them. We say that G is a (*geometric*) t -spanner for P if $d_G(p, q) \leq t \cdot |pq|$ for all pairs of points $p, q \in P$, where $|pq|$ denotes the length of the segment pq . The *dilation*, or *stretch factor*, of G is the minimum t for which G is a t -spanner. Since their introduction by Chew [3] and by Peleg and Ullman [18]—in the latter paper, spanners were defined in a more general graph-theoretic setting—about two decades ago, spanners have been studied extensively: numerous papers on the topic have been written, including several surveys [8, 11, 19], and just recently a book devoted solely to geometric spanners was published [17].

The spanner concept captures the notion of “good” networks when short connections between the points are important. The main question is whether spanners exist that have a small stretch factor and a small number of edges. Ideally, one would like to be able to construct, for any given constant $\varepsilon > 0$, a spanner of stretch factor $1 + \varepsilon$ and with $O(n)$ edges (where the constant in the $O(n)$ bound on number of edges will depend on ε). Other desirable properties of a spanner are for example that the total weight of the edges is small, or that the maximum degree is low. As it turns out, such spanners do indeed exist and they can be constructed in $O(n \log n)$: for any set P of n points in the plane there exists a $(1 + \varepsilon)$ -spanner with $O(n/\varepsilon^2)$ edges, bounded degree, and

*MA was supported by the Netherlands’ Organisation for Scientific Research (NWO) under project no. 612.065.307. MdB was supported by the Netherlands’ Organisation for Scientific Research (NWO) under project no 639.023.301.

[†]Department of Computing Science, TU Eindhoven. PO Box 513, 5600 MB Eindhoven, the Netherlands.

[‡]National ICT Australia Ltd. NICTA is funded through the Australian Government’s Backing Australia’s Ability initiative, in part through the Australian Research Council.

whose total weight is $O(wt(MST(P)))$, where $wt(MST(P))$ is the weight of a minimum spanning tree of P [12, 17]. Other variants consider spanners for points among obstacles [7], fault-tolerant spanners [1, 6] or plane spanners [3].

Despite the large number of papers dealing with geometric spanners, two fundamental problems remained unsolved: how to maintain $(1 + \varepsilon)$ -spanners dynamically and how to maintain $(1 + \varepsilon)$ -spanners kinetically.

In the dynamic setting, one wants to maintain the spanner under insertions and deletions of points in P . The dynamic case has been studied intensively the last few years, but none of the solutions achieved polylogarithmic update time for both insertions and deletions. Very recently Gottlieb and Roditty [10] obtained a breakthrough in this area by presenting a $(1 + \varepsilon)$ -spanner of size $O(n/\varepsilon^d)$ for points in \mathbb{R}^d with $O(\log^3 n)$ update time. (Their update time is only amortized, however, so the problem is still not fully resolved.)

Unfortunately, the kinetic setting is much further from being resolved. In this setting, the points in P move continuously. This means that at certain moments in time the spanner needs to be updated, in order to guarantee it remains a $(1 + \varepsilon)$ spanner at all times. Following standard terminology for kinetic data structures [13, 14] we call these moments *events*. The goal now is to design the spanner in such a way that the number of events is small (under certain assumptions on the point trajectories). Ideally, the number of events should be close to the minimum number of events needed to maintain the structure of interest. For spanners this means the goal is $O(n^2)$ events, as there are sets of n linearly moving points in the plane for which any t -spanner must process $\Omega(n^2/t^2)$ events [9]. Moreover, the *response time*—the time needed for an update when an event happens—should be small, ideally polylogarithmic.

So far the only result on maintaining $(1 + \varepsilon)$ -spanners of moving points in \mathbb{R}^d has been obtained by Gao *et al.* [9]. Their spanner has size $O(n/\varepsilon^d)$. Unfortunately, both the number of events and the response time depend on the *spread* of P , which is the ratio ρ of the maximum pairwise distance to the minimum pairwise distance of points in P : the number of changes is $O(n^2 \log \rho)$ (assuming the points follow low-degree algebraic trajectories) and the time needed to update the spanner at such an event is $O(\log \rho/\varepsilon^d)$.

For points in the plane, one possible alternative is to maintain the Delaunay triangulation of the point set in the L_1 -metric. This gives a spanner that undergoes $O(n^2 \cdot \alpha(n))$ events for linearly moving points, as shown by Chew [4]. However, the stretch factor of the spanner is $\sqrt{10}$ [3]. One may also use the Delaunay triangulation in the Euclidean metric, giving a 2.42-spanner. The best known bound on the number of events for the (Euclidean) Delaunay triangulation is $O(n^2 \cdot \lambda_s(n))$ [2], assuming that the trajectories are described by polynomials of maximum degree s . (Here $\lambda_s(n)$ denotes the maximum length of a (n, s) -Davenport-Schinzel sequence.) The latter result can be extended to environments with obstacles [16].

Thus the main question for kinetic spanners is still unanswered: is there a kinetic $(1 + \varepsilon)$ -spanner with close to a quadratic number of events and polylogarithmic response time?

We answer this question for points in the plane: we present a kinetic $(1 + \varepsilon)$ -spanner of size $O(n/\varepsilon^2)$ that processes $O((n/\varepsilon^2) \cdot \lambda_{s+2}(n))$ events in the worst case, assuming the trajectories of the points can be described by polynomials whose degrees are at most s , and that has only $O(1)$ response time. (The response time does not include the $O(\log n)$ event-scheduling time which is always needed in a kinetic data structure.) Our result uses a new spanner, which we call the *diamond-Delaunay spanner*, which is surprisingly simple in its construction, its analysis, and its kinetic maintenance.

2 The spanner

Let P be a set of n points in the plane, and let ε be a given positive constant. We will first present our new algorithm to construct a spanner for P , then we analyze its spanning ratio and size, and finally we will show how to maintain the spanner when the points in P move continuously.

2.1 The construction

Our spanner construction is quite simple. We define a collection \mathcal{D} of $O(1/\varepsilon^2)$ thin diamonds, which are rotated copies of some fixed diamond Δ_0 . Each diamond induces a convex distance function, and we compute the Delaunay triangulation of the point set P with respect to this distance function. The spanner is then the union of these Delaunay triangulations. Next we make this idea precise.

We first define the diamond Δ_0 . Fix an angle ϕ ; later we will see how to choose ϕ as a function of ε to get a $(1 + \varepsilon)$ -spanner. Now Δ_0 is the diamond centered at the origin whose diagonals are parallel to the x - and the y -axis, respectively, whose leftmost and rightmost vertices have angle ϕ , and whose horizontal diagonal has length 2. We call the longer diagonal of Δ_0 —which will be the horizontal one—its *main diagonal*.

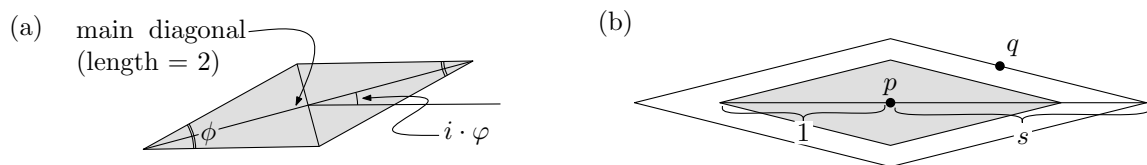


Figure 1: (a) The rotated diamond Δ_i . (b) Two points p and q with $d_0(p, q) = s$.

Next we define the collection \mathcal{D} of rotated diamonds. Let $\varphi := 2\pi/k$, where k is an integer; the exact value of φ depends on ε and will be determined in Section 2.2. For an integer $i > 0$ we let Δ_i denote the diamond Δ_0 rotated around the origin over an angle $i \cdot \varphi$ in counterclockwise direction—see Fig. 1(a). We set $\mathcal{D} := \{\Delta_i : 0 \leq i < k\}$.

Each diamond $\Delta_i \in \mathcal{D}$ defines a convex distance function $d_i : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ in the usual way and as explained next. Consider two points $p, q \in \mathbb{R}^2$. Translate Δ_i such that its center is p , and consider scaling Δ_i with respect to p . Then $d_i(p, q)$ is equal to the scale factor for which q is on the boundary of Δ_i —see Fig. 1(b). Since Δ_i is centrally symmetric, d_i is indeed a distance function [15]. Note that Δ_i plays the role of the unit disk for the distance function d_i . We will call any translated and scaled copy of Δ_i an *i -diamond*.

Because d_i is a distance function, we can construct Voronoi diagrams and Delaunay triangulations based on it [5]. We denote the Delaunay triangulation¹ of the point set P based on d_i by $\mathcal{DT}_i(P)$. Thus for two points $p, q \in P$ the edge (p, q) is an edge in $\mathcal{DT}_i(P)$ if and only if there is an i -diamond with p and q on its boundary that does not contain any other point from P (either inside or on its boundary). Our spanner $\mathcal{S}(P)$ is the union of the Delaunay triangulations \mathcal{DT}_i for $0 \leq i < k$. In other words, (p, q) is an edge in $\mathcal{S}(P)$ if and only if it is an edge in \mathcal{DT}_i for some $0 \leq i < k$. We call $\mathcal{S}(P)$ the *diamond-Delaunay spanner* of P .

2.2 Analysis of the spanning ratio

Next we prove that $\mathcal{S}(P)$ is a $(1 + \varepsilon)$ -spanner for P if we choose ϕ and φ in such a way that $(\cos \phi - \sin \phi) \geq 1/(1 + \varepsilon)$ and $(\sin(\phi/2)/\sin(\varphi/2)) \geq 1 + 2/\varepsilon$. This can be achieved by setting $\phi := \arcsin \frac{\varepsilon}{2(1+\varepsilon)}$ and $\varphi := 2 \arcsin \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)}$. Our proof is based on the following lemma.

Lemma 1 *Let Δ be a 0-diamond. Let p be the leftmost vertex of Δ and let q be a point on the boundary of Δ such that the angle that pq makes with Δ 's main diagonal is at most $\varphi/2$. Then for any point $r \in \Delta$ we have*

$$|pr| + (1 + \varepsilon) \cdot |rq| \leq (1 + \varepsilon) \cdot |pq|.$$

¹Since d_i is not the Euclidean distance, the Delaunay triangulation is not a triangulation of the convex hull but of a different “hull”—see also Section 2.3. Moreover, like for the Euclidean distance function the bounded faces need not be triangles in degenerate cases.

Proof. If $r_x \leq q_x$ then we have a cone with apex p and opening angle ϕ containing q and r , with r being closer to p than q —see Fig. 2(a)—and the analysis becomes similar to the analysis for θ -graphs. Because $|pq|$ is the longest edge in the triangle pqr , we have

$$|rq| \leq (|pq| - (\cos \angle qpr - \sin \angle qpr)|pr|).$$

Hence,

$$|pr| + (1 + \varepsilon) \cdot |rq| \leq |pr| + (1 + \varepsilon) \cdot (|pq| - (\cos \phi - \sin \phi)|pr|) \leq (1 + \varepsilon) \cdot |pq|.$$

If $r_x > q_x$ we can argue as follows. Let z be the rightmost vertex of Δ . Note that $\angle qpz \leq \varphi/2$ —see Fig. 2(b). Hence,

$$\begin{aligned} |pr| + (1 + \varepsilon) \cdot |rq| &\leq |pz| + (1 + \varepsilon) \cdot |zq| \\ &\leq |pq| \cdot (1 + \sin(\varphi/2)/\sin(\phi/2) + (1 + \varepsilon) \sin(\varphi/2)/\sin(\phi/2)) \\ &\leq (1 + \varepsilon) \cdot |pq|. \end{aligned}$$

□

Define a $(1 + \varepsilon)$ -path from a point $p \in P$ to a point $q \in P$ as a path from p to q in $\mathcal{S}(P)$ of length at most $(1 + \varepsilon) \cdot |pq|$. Lemma 1 implies that concatenating the edge pr and a $(1 + \varepsilon)$ -path from r to q yields a $(1 + \varepsilon)$ -path from p to q . We shall use this to show by induction that we have a $(1 + \varepsilon)$ -path between every pair of points.

Theorem 2 *The diamond-Delaunay spanner $\mathcal{S}(P)$ is a $(1 + \varepsilon)$ -spanner for P of size $O(n/\varepsilon^2)$.*

Proof. To bound the stretch factor of $\mathcal{S}(P)$, we must show that for any pair of points $(p, q) \in P \times P$ there is a $(1 + \varepsilon)$ -path from p to q in $\mathcal{S}(P)$. We prove this by induction on $|pq|$. For a pair (p, q) , we let $\alpha(p, q)$ denote the angle that \vec{pq} makes with the positive x -axis.

Base case: Let (p, q) be a closest pair in P . Let $i \in \{0, \dots, k - 1\}$ be such that $|\alpha(p, q) - i \cdot \varphi|$ is minimized. Assume without loss of generality that $i = 0$. Let Δ be the 0-diamond with p as its left vertex and with q on its boundary. Note that q must lie on an edge of Δ not incident to p , which implies that $|qr| < |pq|$ for any point $r \in \Delta$. Since (p, q) is a closest pair, this means Δ cannot contain any other point from P . Hence, (p, q) is an edge in $\mathcal{DT}_0(P)$. This proves the base case.

Induction step: Consider a pair (p, q) that is not a closest pair. The induction hypothesis then states that for every pair (s, t) in P with $|st| < |pq|$ there exists a path in $\mathcal{S}(P)$ of length at most $(1 + \varepsilon) \cdot |st|$. As in the base case, we assume without loss of generality that $|\alpha(p, q) - i \cdot \varphi|$ is minimized for $i = 0$.

Place a small 0-diamond Δ with its leftmost vertex at p that does not contain any other point from P . Grow Δ while keeping p as its leftmost vertex (and such that Δ remains a 0-diamond) until Δ hits some point $r \in P$. By construction there is a 0-diamond with p and r on its boundary that does not contain any other point from P . This means that (p, r) is an edge in $\mathcal{DT}_0(P)$ and, hence, in $\mathcal{S}(P)$. (If Δ hits several points simultaneously then p must have an edge to at least one of them, and the argument still goes through.) So if $r = q$ we are done. Otherwise we argue as follows.



Figure 2: Illustration for the proof of Lemma 1.

Continue to grow Δ , until Δ hits q . Note that q lies on an edge of Δ not incident to p and that $r \in \Delta$. Hence, $|rq| < |pq|$. We can therefore apply the induction hypothesis and conclude that we have a $(1 + \varepsilon)$ -path in $\mathcal{S}(P)$ from r to q . Moreover, we have an edge from p to r in $\mathcal{S}(P)$. Since $|\alpha(p, q) - i \cdot \varphi|$ is minimized for $i = 0$, the vector \overrightarrow{pq} makes an angle of at most $\varphi/2$ with the main diagonal. Hence, we can apply Lemma 1. This finishes the proof that the stretch factor is $1 + \varepsilon$.

To bound the size of $\mathcal{S}(P)$ we note that each $\mathcal{DT}_i(P)$ is a plane graph and therefore has at most $3n - 6$ edges. Since $|\mathcal{D}| = \frac{2\pi}{\varphi}$ and $\varphi = 2 \arcsin \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)}$, we get $|\mathcal{S}(P)| = O(n/\varepsilon^2)$. \square

2.3 Kinetic maintenance of the spanner

Chew [4] has shown that the number of topological changes in the Delaunay triangulation in the L_1 -metric of linearly moving points is $O(n^2\alpha(n))$. As already observed by Chew, the same bound holds for other convex polygonal distance functions such as our distance functions d_i . (In fact, d_i is equivalent to the L_1 -metric after applying an appropriate transformation.) His proof also applies when the points follow algebraic trajectories of maximum degree s . The bound on the number of events then becomes $O(n\lambda_{s+2}(n))$. Since the diamond-Delaunay spanner consists of $O(1/\varepsilon^2)$ diamond-Delaunay triangulations, the number of events for the spanner is $O((n/\varepsilon^2) \cdot \lambda_{s+2}(n))$.

To maintain the spanner when the points move we only have to maintain the Delaunay triangulations $\mathcal{DT}_i(P)$. As already observed by Karavelas and Guibas [16], maintaining Delaunay triangulations is quite easy; the fact that our Delaunay triangulations are defined with respect to the convex distance functions d_i does not change much. For completeness we describe below how the maintenance is done. For simplicity we assume that degeneracies only occur at events. In particular, we assume that in between events there are no four points in P on the boundary of a common i -diamond.

When using the normal Euclidean distance, the Delaunay triangulation of a point set P is a triangulation of the convex hull of P . For $\mathcal{DT}_i(P)$ something similar holds, except that now the notion of a hull is different, as explained next.

Each vertex v of an i -diamond Δ defines a cone with apex v , by extending the two edges incident to v to half-lines. We call such cones i -cones—see Fig. 3(a). Now a point $p \in P$ is a vertex of the d_i -hull of P if and only if there is an i -cone with apex p that does not contain any other point from P , and a pair of points (p, q) forms a *hull edge* if and only if there is an i -cone with p and q on its boundary that does not contain any other point from P . The d_i -hull of P is connected and outerplanar (that is, every vertex of the d_i -hull is on the outer face). Hence, the d_i -hull partitions the plane into one unbounded face, and a set of bounded faces. Each bounded face is bounded by a simple cycle. The Delaunay triangulation $\mathcal{DT}_i(P)$ consists of the d_i -hull and a triangulation of its bounded faces by *interior edges*—see Fig. 3(b).

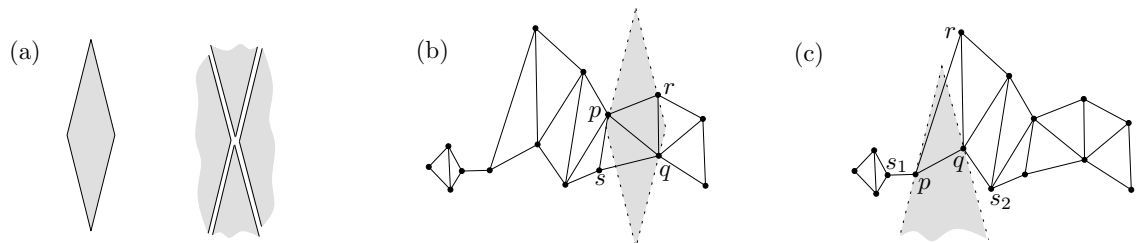


Figure 3: (a) shows an i -diamond and the four types of cones defined by it. (b) and (c) show a Delaunay triangulation \mathcal{DT}_i , with in (b) showing the empty diamond certifying a hull edge and (c) showing the empty cone certifying a hull edge.

Now suppose the points start to move. When does the Delaunay triangulation $\mathcal{DT}_i(P)$ change? To detect such changes we have two types of certificates, one for the hull edges and one for the interior edges of $\mathcal{DT}_i(P)$. Next we describe these two certificates, and discuss what needs to be done when they fail.

Interior-edge certificates: In between events, every interior edge (p, q) is incident to two triangles. Let r and s be the two other vertices of these two triangles. We maintain an **InDiamond** certificate certifying that s is outside the unique i -diamond passing through p, q, r , as shown in Fig. 3(b). When this certificate fails we perform an edge flip: we replace (p, q) by (r, s) .

Hull-edge certificates: Every hull-edge (p, q) is incident to at most one triangle and incident to at most four other hull edges. Let r be the third vertex of this triangle (if it exists) and let s_1, \dots, s_k be the other endpoints of the hull edges incident to (p, q) , where $k \leq 4$ denote the number of incident hull edges—see Fig. 3(c). (It could be that r is one of the s_i .) We maintain at most five **InCone** certificates certifying that r and s_1, \dots, s_k are outside the unique i -cone passing through p and q and not containing any other points. When an **InCone** certificate fails² we either insert or delete a vertex from the d_i -hull. Note that when this happens we should also delete or insert an edge into the triangulation.

Thus, whenever an interior-edge certificate or a hull-edge certificate fails we can update the spanner in $O(1)$ time. Since we also have to schedule the failure times of the events in an event queue to be able to handle them in the correct chronological order, the total event handling time is $O(\log n)$. We get the following theorem.

Theorem 3 *The diamond-Delaunay spanner of a set of n points in the plane can be maintained kinetically in $O(1)$ time per event, plus $O(\log n)$ time to update the event queue. When the points follow algebraic trajectories of maximum degree s , then the number of events is $O(n\lambda_{s+2}(n))$.*

3 Concluding remarks

We have presented a new $(1 + \varepsilon)$ -spanner, the diamond-Delaunay spanner, for a set of points in the plane. Our spanner has size $O(n/\varepsilon^2)$ and can easily be maintained as the points move. The number of events almost matches the $\Omega(n^2)$ lower bound from Gao *et al.* [9], and each event can be handled in $O(1)$ time (plus $O(\log n)$ to update the event queue). This is the first kinetic $(1 + \varepsilon)$ -spanner for which the number of events and the response time do not depend on the spread of the point set.

Unfortunately our approach does not generalize to higher dimensions, because there the Delaunay triangulation can have a quadratic number of edges. Another direction for further research is to develop an efficient and responsive kinetic $(1 + \varepsilon)$ -spanner in the plane whose total weight is $O(wt(MST(P)))$ and/or in which every point has small degree.

References

- [1] M. A. Abam, M. de Berg, M. Farshi and J. Gudmundsson. Region-fault tolerant geometric spanners. In Proc. of the 18th ACM-SIAM Symposium on Discrete Algorithms, pp. 1–10, 2007.
- [2] G. Albers, L. J. Guibas, J. S. B. Mitchell and T. Roos. Voronoi diagrams of moving points. International Journal on Computational Geometry and Applications, 8(3):365–380, 1998.
- [3] L. P. Chew. There is a planar graph almost as good as the complete graph. In Proc. of the 2nd ACM Symposium on Computational Geometry, pp. 169–177, 1986.
- [4] L. P. Chew. Near-quadratic bounds for the L_1 Voronoi diagram of moving points. Computational Geometry: Theory and Applications, 7:73–80, 1997.

²An alternative approach is to maintain two sorted lists of the points in P , one ordered according to the projections onto a line orthogonal to one side of an i -diamond and another ordered according to the projections onto a line orthogonal to a second (non-parallel) side. It is easy to see that every change in d_i -hull corresponds to a change in one of the two sorted lists.

- [5] L. P. Chew and R. L. (Scot) Dyrsdale, III, Voronoi diagrams based on convex distance functions. In Proc. of the 1st Symposium on Computational geometry, pp. 235–244, 1985.
- [6] A. Czumaj and H. Zhao. Fault-tolerant geometric spanners. In Proc. of the 19th ACM Symposium on Computational Geometry, pp. 1–10, 2003.
- [7] G. Das. The visibility graph contains a bounded degree spanner. In Proc. of the 9th Canadian Conference on Computational Geometry, 1997.
- [8] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, eds., Handbook of Computational Geometry, Elsevier Science Publishers, pp. 425–461, 2000.
- [9] J. Gao, L. J. Guibas and A. Nguyen. Deformable spanners and applications. Computational Geometry: Theory and Applications, 35(1-2):2-19, 2006.
- [10] L.-A. Gottlieb and L. Roditty. Improved algorithms for fully dynamic geometric spanners In Proc. of the 18th ACM-SIAM Symposium on Discrete Algorithms, 2008.
- [11] J. Gudmundsson and C. Knauer. Dilation and detour in geometric networks. In Teofilo Gonzalez, ed., Handbook on approximation algorithms and metaheuristics. Chapter 52. Chapman & Hall/CRC, 2006.
- [12] J. Gudmundsson, C. Levcopoulos and G. Narasimhan. Improved greedy algorithms for constructing sparse geometric spanners. SIAM Journal on Computing, 31(5):1479–1500, 2002.
- [13] L. J. Guibas. Kinetic data structures: A state of the art report. In Proc. 3rd Workshop on Algorithmic Foundations of Robotics, pp. 191-209, 1998.
- [14] L. J. Guibas. Motion. In J. Goodman and J. O'Rourke, eds., Handbook of Discrete and Computational Geometry, pp. 1117-1134. CRC Press, 2nd edition, 2004
- [15] Handbook of Convex Geometry, Vol. A, P. M. Gruber & J. M. Wills, eds., North Holland, Amsterdam-London-New York-Tokyo, 1993.
- [16] M. I. Karavelas and L. J. Guibas. Static and kinetic geometric spanners with applications. In Proc. of the 12th Symposium on Discrete Algorithms, pp. 168–176, 2001.
- [17] G. Narasimhan and M. Smid. Geometric spanner networks. Cambridge University Press, 2007.
- [18] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. In Proc. of the 6th ACM Symposium on Principles of Distributed Computing, pp. 77–85, 1987.
- [19] J. Soares. Graph spanners: A survey. Congressus Numerantium, 89:225–238, 1992.